ENABLING DEMAND RESPONSE PARTICIPATION FOR HPC SYSTEM

Kishwar Ahmed, Jason Liu
Florida International University, Miami, FL, USA

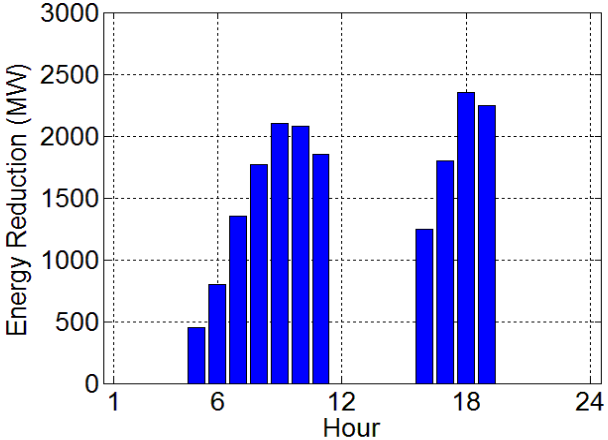
Xingfu Wu
Argonne National Laboratory, Argonne, IL, USA

Outline

- Motivation
  - Why demand response is important?
  - HPC system as demand response participant?
- Model and Simulator
  - How we model HPC demand response participation?
  - How we simulate the proposed model?
- Experiment
  - How we compare our model with existing solutions?
- Conclusions
  - Some ongoing and future works

What is Demand Response (DR)?

- Participants reduce energy consumption
  - During transient surge in power demand
  - Other emergency events
- A DR example:
  - Extreme cold in beginning of January 2014
  - Closure of electricity grid
  - Emergency demand response in PJM and ERCOT



Energy reduction target at PJM

Demand Response Getting Popular!

SUSTAINABILITY

Why Apple Is Getting into the Energy Business

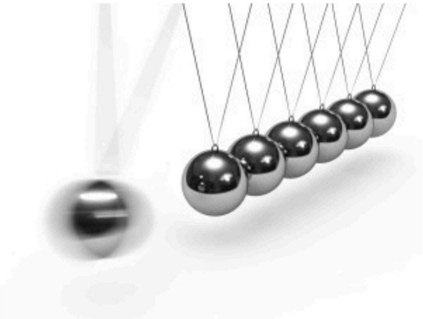
by Peter Fox-Penner

NOVEMBER 25, 2016

But solar electricity is only the beginning of the future energy marketplace. Many companies are already in markets where "demand-response" contracts enable them to sell the right to manage a portion of their power use, allowing them to be paid for reducing their energy during hours when the spot price of power is high. In the future, in addition to selling actual

Equinix 'in R&D phase' of demand response experiments

OCTOBER 16, 2015 BY BRENDAN COYNE — 1 COMMENT



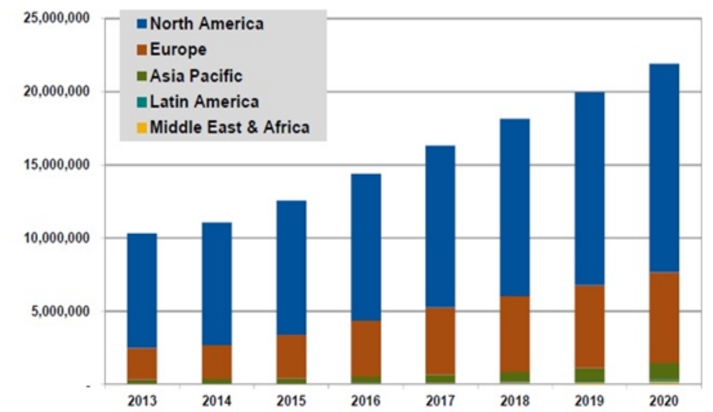
Equinix is "in the R&D phase" of testing demand response technologies, according to UK MD Russell Poole.

National Grid wants more businesses to help balance the electricity system by turning power off or on, or adjusting loads, in return for payment. However, mission critical sites have traditionally been reluctant to increase any perceived risk factors for relatively low rewards. National Grid though, is rethinking its mechanisms and contract structures.

DEMAND RESPONSE WILL DOUBLE BY 2020: HERE'S WHY

BY JESSICA KENNEDY SEPTEMBER 3, 2014

Chart 1.1 Total DR Sites by Region, World Markets: 2013-2020



(Source: Navigant Research)

HPC System as DR Participant?

- HPC system is a major energy-consumer
  - China's 34-petaflop Tianhe-2 consumes 18MWs of power
    - Can supply small town of 20,000 homes
  - The power usage of future HPC system is projected to increase
    - Future exascale supercomputer has 20MWs power limit
    - But not possible with current system architecture
- A number of recent surveys on possibility of supercomputer's participation in DR program
  - Patki et al. (in 2016), Bates et al. (2015)
  - "*Supercomputing systems in the U.S. may be willing to participate in the demand response programs if tighter and more frequent communication can be established*"

Outline

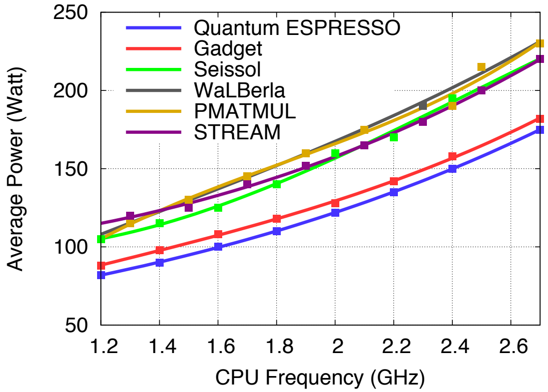
- Motivation
  - Why demand response is important?
  - HPC system as demand response participant?
- Model and Simulator
  - How we model HPC demand response participation?
  - How we simulate the proposed model?
- Experiment
  - How we compare our model with existing solutions?
- Conclusions
  - Some ongoing and future works

Demand Response Model

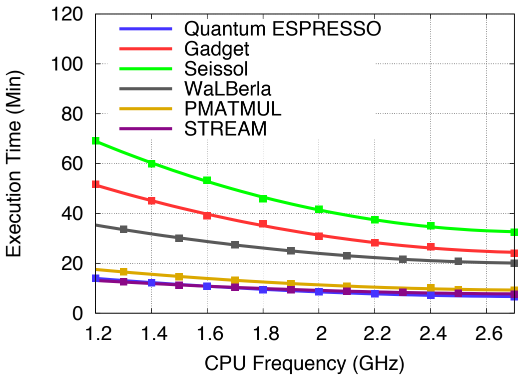
- Power and performance prediction model
  - Based on a polynomial regression model
- Resource provisioning
  - Determine processors' optimal frequency to run the job
- Job scheduling
  - Based on FCFS with possible job eviction (to ensure power bound constraint)

Power and Performance Prediction

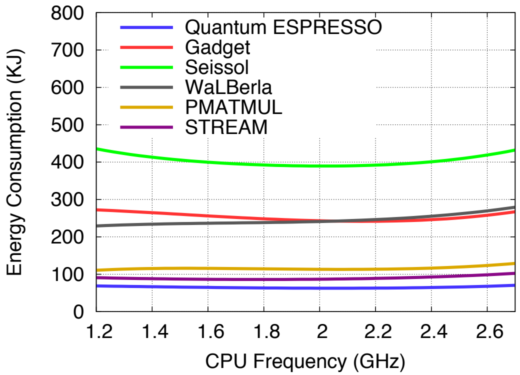
$$p(j,f) = a + b \cdot f + c \cdot f^2 + d \cdot f^3$$



$$t(j,f) = \alpha + \beta \cdot f + \gamma \cdot f^2$$



$$e(j,f) = n_j \cdot p(j,f) \cdot t(j,f)$$



Job Scheduling and Resource Provisioning

Algorithm 1 HPC Demand Response Job Scheduler

```
1: find the first eligible job j in Q
2: if job j exists then
3:     dequeue job j from Q
4:     allocate n_j processors to run job j
5:     R ← R ∪ {j}
6:     goto line 1
7: end if
8: determine optimal frequency ∀j ∈ R (section III-C)
9: if no optimal solution exist then
10:    evict jobs to reduce power consumption (section III-D)
11:    goto line 8
12: end if
13: reset processor frequency if changed ∀j ∈ R
```

Minimize: $\sum_{j \in R} e_R(j, f_j)$

subject to constraints (4) and (5)

$$e_R(j, f_j) = (1 - \alpha_j) \cdot n_j \cdot p(j, f_j) \cdot t(j, f_j)$$

$$f_{min} \leq f_j \leq f_{max} \tag{4}$$

$$p_{run} = \sum_{j \in R} p(j, f_j) \leq \hat{p} \tag{5}$$

Maximize: $\sum_{j \in R} (x_j \cdot e_X(j))$

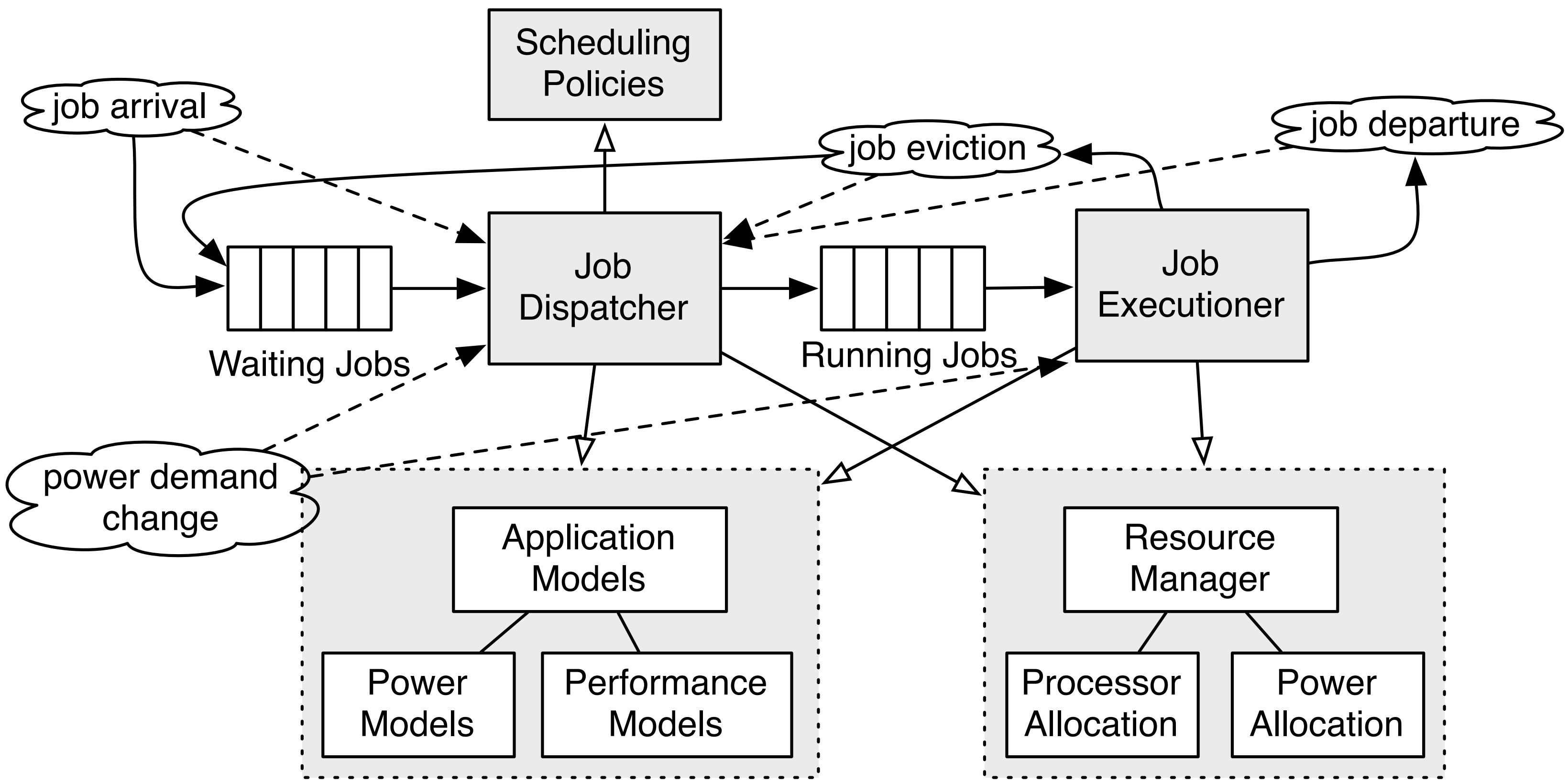
subject to $\sum_{j \in R} (x_j \cdot p(j, f_{min})) \leq \hat{p}$

$$e_X(j) \approx \alpha_j \cdot n_j \cdot p(j, f_j) \cdot t(j, f_j)$$

Job Scheduler Simulator

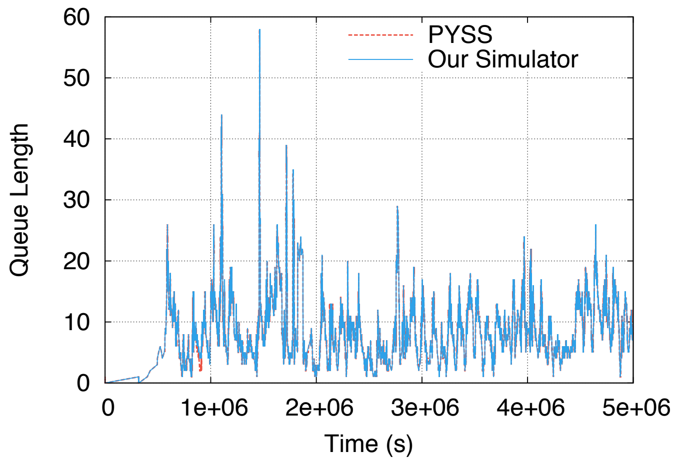
- Developed our scheduler simulator
  - Trace-driven capability
  - Flexibility to incorporate new scheduling functions, power-aware methods, as well as demand response models
- Based on Simian
  - An open-source, process-oriented parallel discrete-event simulation engine
  - Some unique features
    - A minimalistic design (only 500 lines of code base)
    - Takes advantage of just-in-time (JIT) compilation
    - For some models, outperformed simulators using compiled languages such as C or C++
    - Recent significant effort on models based on Simian
      - For example, GPU models (Chapuis et al.), interconnection models (Ahmed et al.)

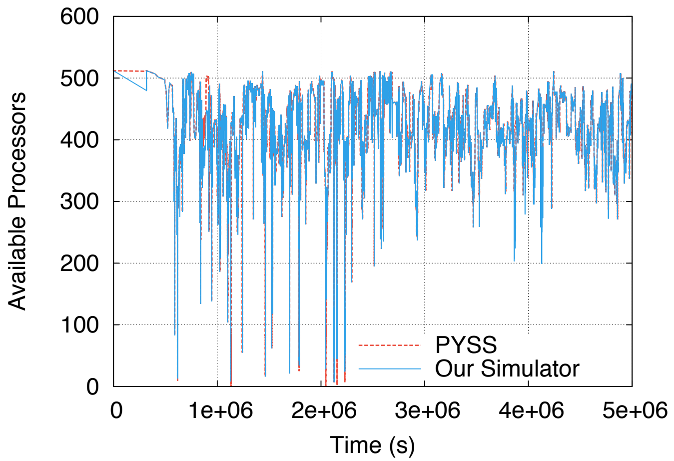
Job Scheduling Simulator (Contd.)



Job Scheduling Simulator (Contd.)

- Validated against PYSS
  - A python-based scheduler simulator for HPC workload
  - Has been used to study various scheduling algorithms in HPC system
- Collected workload trace
  - Parallel Workloads Archive
  - Contains information such as job start time, job run time, number of requested processors, etc.





Outline

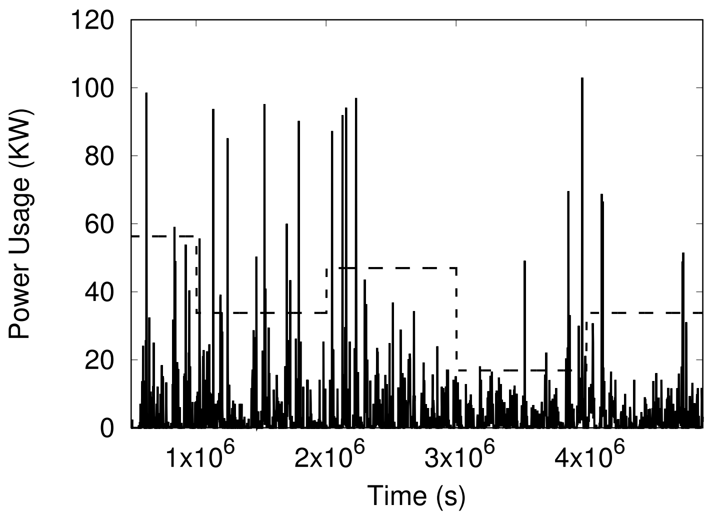
- Motivation
  - Why demand response is important?
  - HPC system as demand response participant?
- Model and Simulator
  - How we model HPC demand response participation?
  - How we simulate the proposed model?
- Experiment
  - How we compare our model with existing solutions?
- Conclusions
  - Some ongoing and future works

Experiment

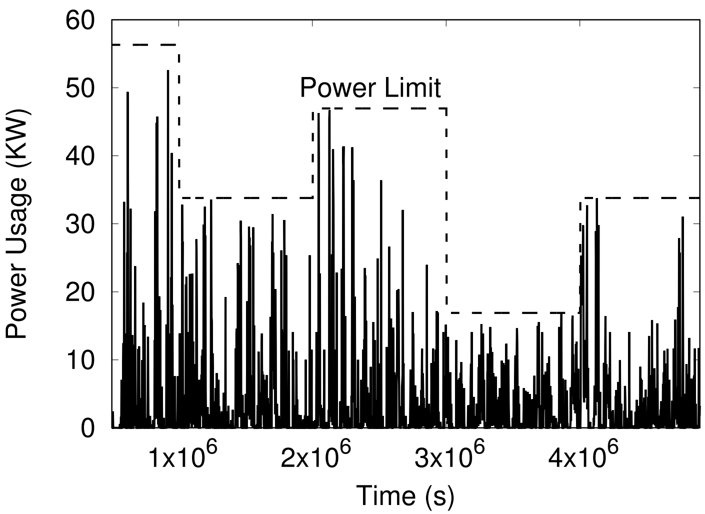
- Workload trace collected from Parallel Workloads Archive
- Power and performance data collected from literature for HPC applications
- Two scheduling policies
  - Used in Linux kernel of Intel processors
  - Performance-policy
    - Always chooses **maximum frequency** to ensure best application runtime
  - Powersave-policy
    - Always chooses the **minimum frequency** to minimize the power consumption

Power Capping

- Power limit 50%, 30%, 41.7%, 15% and 30% of system's peak power



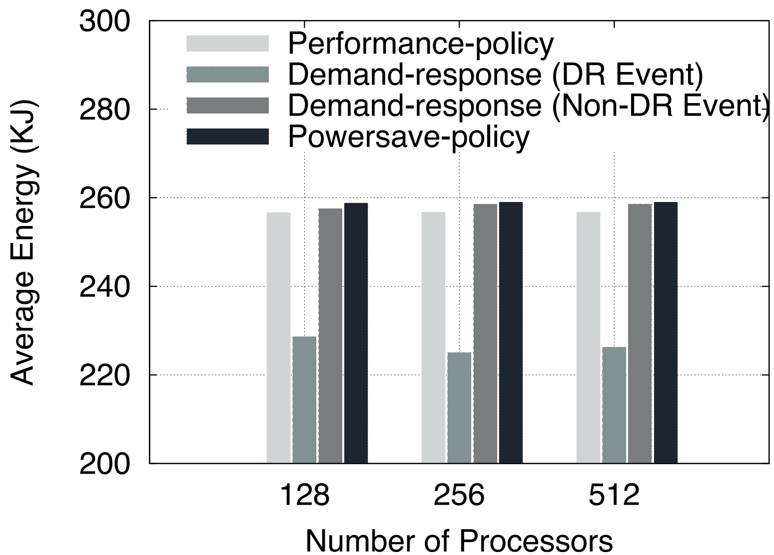
Performance-policy

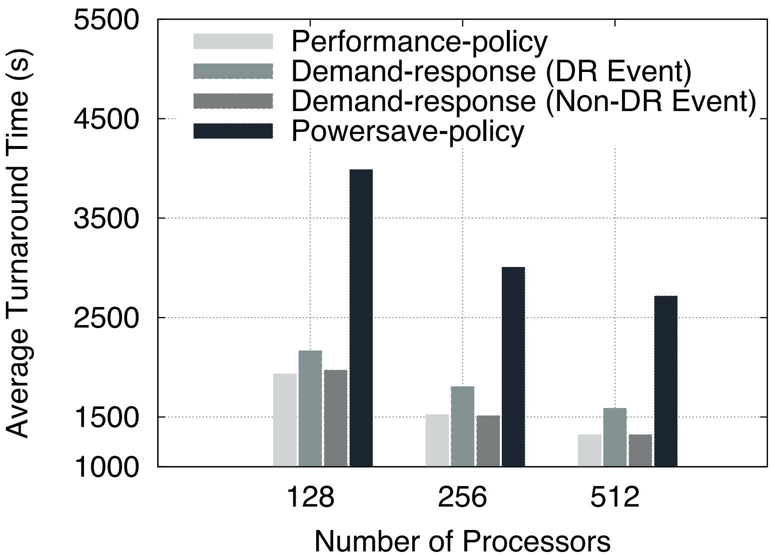


Demand-response

Observation: power usage of demand-response does not go beyond dynamically changed power limit

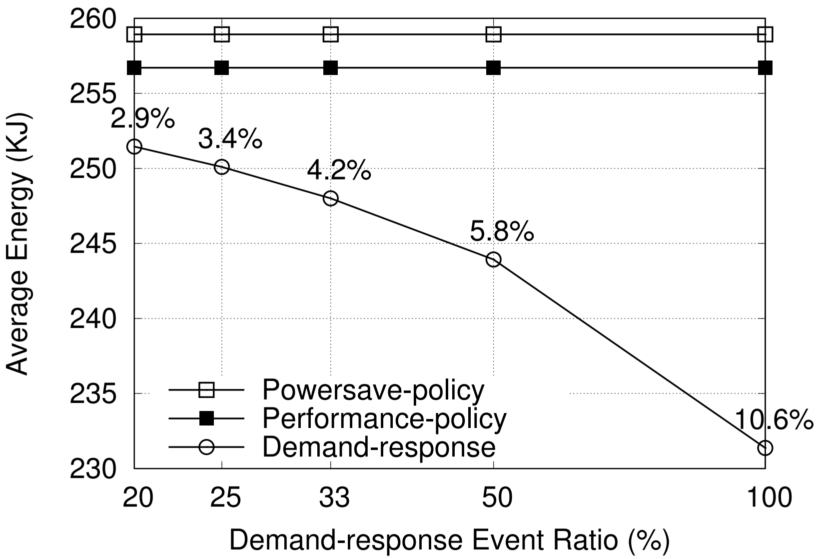
Energy vs. Performance

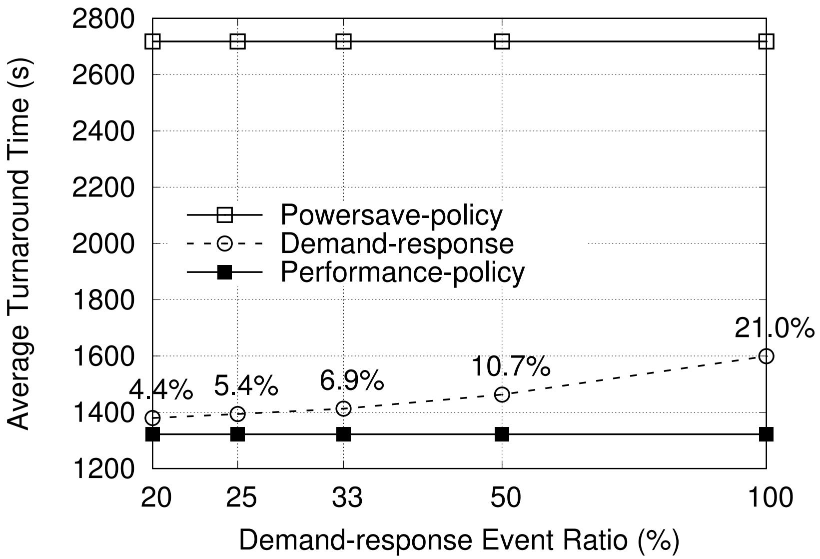




Observation: Reduced energy consumption at moderate increase in turnaround time

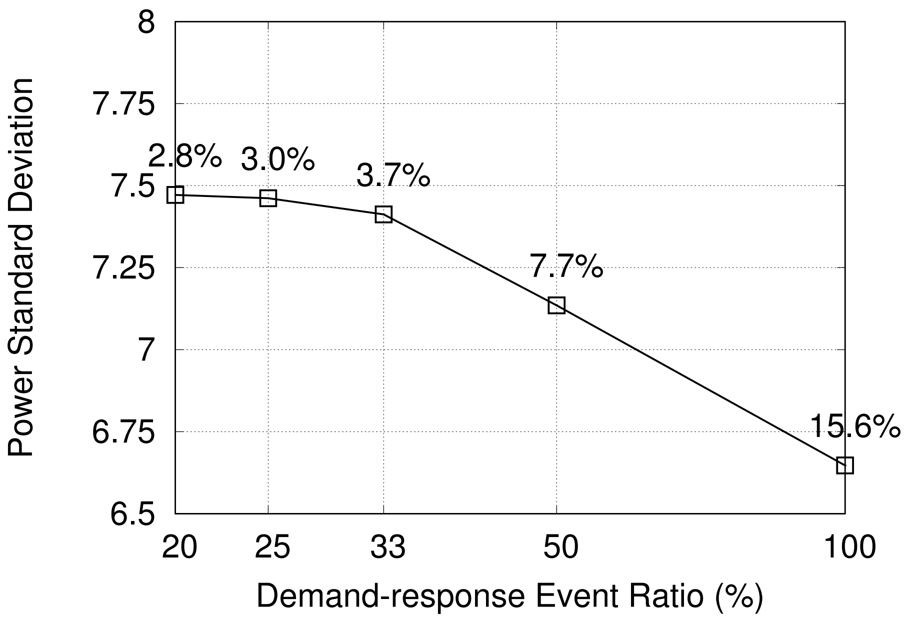
Impact of Demand-response Event Ratio





Observation: Average energy decreases with longer demand response event

Power Stability



Observation: Our algorithm can achieve reduced power standard deviation than Performance-policy; therefore, contributing to stabilize power system

Conclusions

- We studied
  - Possibility of HPC system's demand response participation
- We proposed
  - An HPC job scheduling simulator
  - A demand-response model which ensures
    - Power capping property
    - Demand response participation
    - Energy reduction
- Ongoing and future works
  - Consideration of other major subsystems: cooling, memory
  - Power and performance measurement of HPC applications on real system
  - A prediction model for prediction of unknown HPC applications' characteristics (e.g., power usage)

References

- T. Patki, N. Bates, G. Ghatikar, A. Clausen, S. Klingert, G. Abdulla, and M. Sheikhalishahi, "Supercomputing centers and electricity service providers: a geographically distributed perspective on demand management in Europe and the United States," in International Conference on High Performance Computing. Springer, 2016
- N. Bates, G. Ghatikar, G. Abdulla, G. A. Koenig, S. Bhalachandra, M. Sheikhalishahi, T. Patki, B. Rountree, and S. Poole, "Electrical grid and supercomputing centers: an investigative analysis of emerging opportunities and challenges," *Informatik-Spektrum*, 2015
- G. Chapuis, S. Eidenbenz, N. Santhi, and E. J. Park, "Simian integrated framework for parallel discrete event simulation on GPUs," in *2015 Winter Simulation Conference (WSC)*, Dec 2015, pp. 1127–1138.
- G. Chapuis, D. Nicholaeff, S. Eidenbenz, and R. S. Pavel, "Predicting performance of smoothed particle hydrodynamics codes at large scales," in *Winter Simulation Conference (WSC), 2016*. IEEE, 2016, pp. 1825– 1835
- K. Ahmed, M. Obaida, J. Liu, S. Eidenbenz, N. Santhi, and G. Chapuis, "An integrated interconnection network model for large-scale performance prediction," in *Proceedings of the 2016 annual ACM Conference on SIGSIM Principles of Advanced Discrete Simulation*. ACM, 2016, pp. 177–187.
- K. Ahmed, J. Liu, S. Eidenbenz, and J. Zerr, "Scalable interconnection network models for rapid performance prediction of HPC applications," in *2016 IEEE 18th International Conference on High Performance Computing and Communications (HPCC)*, Dec 2016, pp. 1069–1078

Thank you! Questions?